# From simple digital twin to complex digital twin Part I: A novel modeling method for multi-scale and multi-scenario digital twin

Wenjie Jia, Wei Wang *, Zhenzu Zhang

School of Mechanical and Electrical Engineering,

University of Electronic Science and Technology of China, Chengdu 611731, China

## Abstract

In recent years, the digital twin has attracted widespread attention as an important means of digitalization and intelligence. However, the digital twin is becoming more and more complex due to the expansion of need on the simulation of multi-scale and multi-scenario in reality. The instance of digital twin in references mostly concentrates a particular application, while it is still a lack of a method for constructing the complex digital twin in the total elements, the variable scale of working environments, changeable process, not even the coupling effects. In this paper, a novel modeling method for such a complex digital twin is proposed based on the standardized processing on the model division and assembly. Firstly, the complex model of digital twin is divided into several simple models according to the composition, context. component, and code in 4C architecture. Composition and context make the digital twin focus on the effective elements in a specific scale and scenario. Component and code develop the digital twin in standard-based modularization. Secondly, assemble the simple models of digital twins into the complex model through information fusion, multi-scale association and multi-scenarios iterations. Ontology establishes the complete information library of the entities on different digital twins. Knowledge graph bridges the structure relationship between the different scales of digital twins. The scenario iterations realize the behavior interaction and the accuracy calculation results. It provides an implementable method to construct a complex model of digital twin, and the reuse of components and code also enables rapid development of digital twins.

*Index Terms*—complex digital twin, simple digital twin, digital twin modeling, smart manufacturing

## 1 Introduction

The digital twin was first proposed by Grieves in 2003 [1], which did not arouse widespread attention at that time. However, with the development of hardware and information technology such as computers, sensors, big data, artificial intelligence, and others, the realization of the digital twin has become possible. NASA proposed the concept of the digital twin in 2012 [2]. Subsequently, more and more universities and enterprises have invested in the research of digital twins. From 2017 to 2019, the digital twin had been listed by Gartner as one of the top ten strategic technology trends for three consecutive years. Digital twins can realize the integration of physical space and virtual space, so it is considered an effective way to realize intelligence manufacturing [3].

In the implementation of the digital twin, modeling methods have always been one of the key technologies, and some scholars have made relevant explorations in modeling. Grieves et al. proposed that the digital twin consists of real space, virtual space, and the links for data flow and information flow between real space and virtual space [4]. Tao et al. proposed the five-dimension model of the digital twin, including physical entity, a virtual entity, service system, digital twin data, and connection network, among which virtual entity includes geometry model, physics model, behavior model, and rule model [5]. Zhuang et al. explored the concept of the digital twin and the implementation approach of product digital twin in different stages of the life cycle [6]. Alam et al. proposed a digital twin reference model consisting of physical things, cyber things and hybrid things [7]. These reference models provide the basic structure of the digital twin.

In recent years, the digital twin had been widely applied to various fields, including aerospace [8,9], machinery manufacturing [10,11], medical care [12,13], ship [14], and urban construction [15,16]. More and more scholars have done further research on the application of digital twins. Zhao et al. developed the prototype system of a digital twin workshop based on the Unity3D platform to realize the visual monitoring of real-time data [17]. Ritto et al. use a physics-based computational model and a machine learning classifier to detect damage in structures [18]. Lv et al. proposed a digital twin-driven human-robot collaborative assembly framework based on image recognition and deep reinforcement learning technologies to improve assembly efficiency and safety [19]. Lai et al. used an improved finite element method and an analytical model to monitor the strain, stress, and load of a boom crane [20]. In brief, the pattern of realization of digital twins can be summarized into the following steps, determine application contexts, analyze functional requirements, develop the digital twin (including hardware deployment and software development), commissioning, and implementation. And the digital twin in the literature only simulates the behavior in one or a limited scenario, which can be thought of as the **simple digital twin**. Furthermore, even some digital twins have similar elements, there are different ways to develop the components of a digital twin. And the actual situation is indeed the case, in the process of building these specific models, the tools and platforms used by developers are different. This leads to the difficulty of these models interacting with each other.

Currently, in the face of multi-scale, multi-scenario, multi-dimensional applications of digital twin, here we call it **complex digital twin**, it is not easy to start the modeling of complex digital twin on the whole element, the whole process, and the whole business in one time. Partly because achieving the modeling in full scale from macro composition to micro composition in current modeling level and computing power is virtually impossible, let alone multi-scenario, multi-dimensional applications. Some scholars are trying to construct a new standard method on the modeling of the complex digital twin to simulate, predict and optimize behavior in multi-scale and multi-context. Rosen et al. proposed the way to achieve a digital twin by Modularity-Autonomy-Connectivity-Digital Twin [21]. Wang et al. proposed a new industrial internet model based on hierarchical multi-granularity digital twin, different types of models have different perceptions and control objects, and the same-layer models cooperate to complete production tasks under the control of the high-layer model [22]. Pan et al. divided the production logistics system into three levels, and used the cloud-fog-edge-based control framework for real-time monitoring, decision and control [23]. However, these articles simply proposed to divide the digital twin into different layers, the interfaces of digital twins are still lacking. Considering that software is the ultimate carrier of digital twins, the development processing of the complex digital twin is unclear.

Another question is rising how to integrate, interact, expand, and reuse the modeling of the digital twins. The emergence of information technology opened a gate of value-added innovative services. Ontology provides a semantic model that computers can understand, which can integrate multiple information of complex models. Bao et al. use the ontology-based model to describe the resource and process for the assembly workshop [24]. Dai et al. divided the as-fabricated data into geometric information and non-geometric information and build the ontology-based information model of as-fabricated parts [25]. Singh et al. built an ontology model and proposed an overall methodology including a map, define, create, convert and populate to manage databases for digital twins [26]. Lu et al. created digital twins for all the manufacturing resources of an international company by developing ontology [27]. The knowledge graph is usually used to describe the attributes of physical entities and their relations of them. Liu et al. build a quality knowledge model based on the knowledge graph to express product quality factors from three different scales [28]. Sun et al. used the assembly-commissioning task knowledge graph to extract the action sequence [29]. The ontology and knowledge graph can be used to reorganize and call the modules, which also provide the possibility for the construction of scalable complex digital twins.

In a word, there are two articles, Part I and Part II study the modeling and application of complex digital twins. In this paper, a novel modeling method for a complex digital twin is proposed in part I, the rest of this article is organized as follows. In Section 2, the characteristics of a complex digital twin are analyzed, and the modeling process is given. Section 3 discusses how to divide a complex digital twin into several simple digital twins through the division of layers and contexts, how to build simple digital twins through the division of functional components. Then in Section 4, the ontology model and knowledge graph are introduced to assemble simple digital twins into a complex digital twin, including the information fusion, the multi-scale association, and the multi-context interaction. Section 5 is the conclusion of this study. The implementation of a complex digital twin, taken a multi-scale and multi-scenario workshop as an example, will illustrate in part II.

## 2 Characteristics and modeling architecture of the complex digital twin

### 2.1 General characteristics of the complex digital twin

A digital twin is the mapping of physical entities in virtual space. The characteristics of the digital twin could be considered as making it more reflect the reality. Schleich et al. proposed four important characteristics of the idea reference of digital twins: scalability, interoperability, expansibility, and fidelity [30]. Scalability is to provide insight at different scales. Interoperability is the ability to convert, combine, and establish equivalence between different model representations. Expansibility could integrate, add, or replace digital twins in a quick time. Fidelity means very close to the physical entities.

Therefore, the complex digital twin has a connotation that has more functions, more scales of composition, more dimensional data, more scenarios, and so on. The spatial span of the complex digital twins may be huge, and it is impossible to express all the information on the same scale, so the scalability of complex digital twins is the ability to present the correct data and hide the irrelevant data on a corresponding scale. The complex digital twins may compose of many simple models of digital twins, the interoperability of complex digital twins is not only the ability to convert, and to combine between different models representations in the same objects, but also the ability to convert, and to combine models of different objects. The expansibility of complex digital twins is the ability to integrate, add, or replace the model of different objects. Limited by current modeling methods and computing power, it is very hard to construct a complex digital twin completely equivalent to a physical entity. Therefore, the fidelity of complex digital twins is the ability to keep as close as possible to the physical product in the required functions or behaviors.

## 2.2 The special characteristics of a complex digital twin shop floor

Take a digital twin shop floor as an example to further illustrate the characteristics of a complex digital twin. The shop floor is a multi-scale system, and the data of its digital twins usually need to be presented at different scales. For example, the key data usually include production scheduling, product qualification rate, and energy consumption when the research object is the entire workshop. When the research object is a machine tool equipment, the key data will change to the spindle speed, feed rate, motor current. So the scales (the layers) of the shop floor could include the production, the unit, and the equipment.

In addition, the different layers in the shop floor also include multiple contexts. For example, the shop floor layer includes the contexts such as production scheduling, power consumption statistics, and the equipment layer concludes the contexts such as data monitoring, remote control, and fault diagnosis. As shown in Figure 1, the lots of scales and context made of a shop floor, it is indeed a complex system, and its digital twins should have the following characteristics:

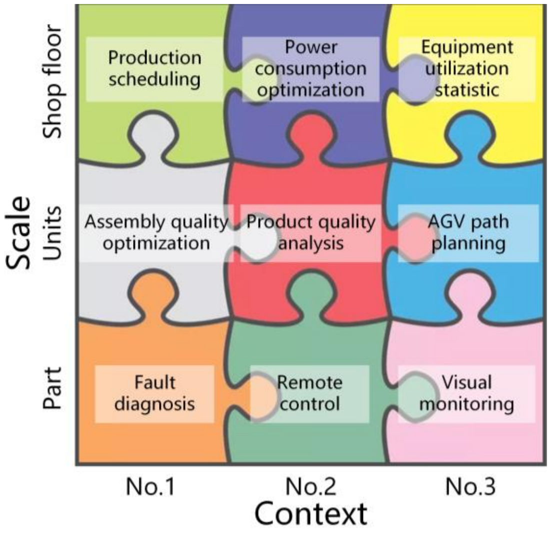


Figure 1. Complex digital twin shop floor

(1) **Scalability:** Scalability means that the content and data are expressed in complex digital twin shop floor could adaptively change as the research object changes. When the research object is the shop floor, the digital twin shows the production schedule, production efficiency, and others. When the research object is a production line, the digital twin shows the status of machine tools, the position of AGV, and so on. When research is focused on certain equipment, the digital twin shows the actions, health status, and others.

(2) **Interoperability:** The digital twin shop floor contains virtual models of many objects, and each virtual model may be composed of several models, such as geometric models, physical property, behavior models and rules to realize complex functions. Interoperability means the model could interact with each other, and these models can belong to the same object or different objects. For example, the data of the machine tool behavior model can drive the movement of the geometric

model, and the interaction between the behavior model of AGV and the behavior model of the robotic arm can realize the cooperation of them.

(3) **Expansibility:** On the shop floor, the number, type and even the location layout of equipment may be adjusted along with the overall planning. Therefore, expansibility means that when the physical space or functional requirements change, the digital twin can be reconfigured through simple operations to maintain synchronization with the physical entity.

(4) **Fidelity:** Although it is impossible to make the virtual models the same as the physical entities at the current stage, some methods can be used to improve the fidelity of the model as much as possible, such as constructing a model that considers more influencing factors, using more advanced algorithms and others.

### 2.3 Modeling process of the complex digital twin

From the above analysis, it is difficult to directly construct a complex digital twin in one time partly because it covers different scales and contexts. Therefore, modeling ideas based on division-assemble is proposed to build a complex digital twin, as shown in Figure 2. First, a complex digital twin is divided into several simple and implementable digital twins. The first problem to be solved is how to divide a complex model. Since the complex digital twin covers different scales and contexts, it can be divided into different layers according to the scales, and then divided into different contexts. Then these simple digital twins are assembled into a complex digital twin which has a more complex function. How to assemble it is an another important question since these simple digital twins may be independent of each other. Here, ontology model can provide a suitable container to fuse the related information of the research object. The knowledge graph can describe the attributes and relations of physical entities, so it is an effective tool for describing the scale associations of simple digital twins. The interaction of contexts are also the key issues that need to be solved, it can realize through the behavior interaction and calculation iteration between the simple digital twins.

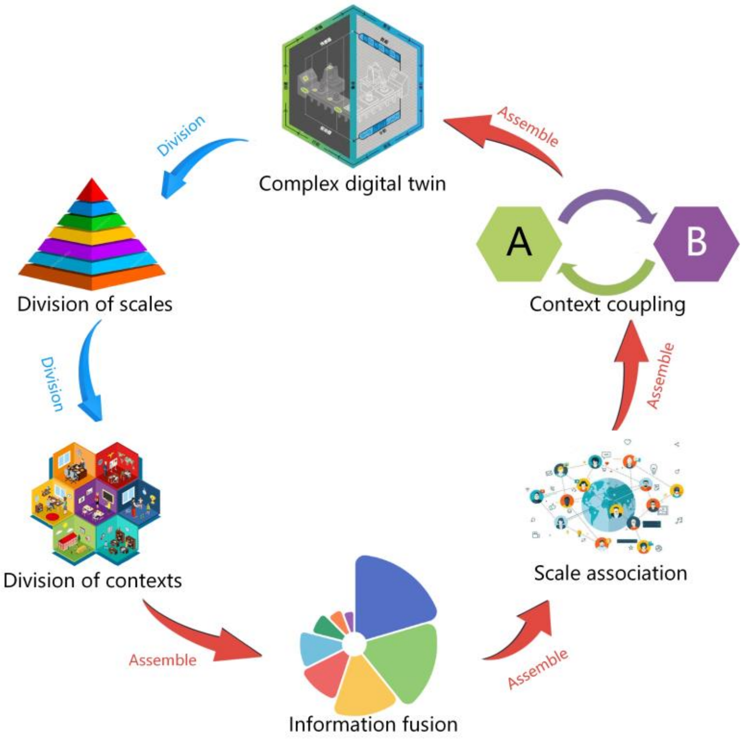


Figure 2. Modeling process of complex digital twins

## 3 Division of the complex digital twin in 4C architecture

$$M_{DT} = (Composition, Context, Component, Code) \tag{3-1}$$

This paper proposes a 4C architecture to the division of the complex digital twin as shown in (3-1), it is a hierarchical division method. 4C architecture contains composition, context, component, and code, composition divides the digital twin into different layers (scales), context is the specific application scenario, the component is the functional unit for building simple digital twins, code is the specific implementation of components. The detailed structure of the 4C architecture is shown in Figure 3.

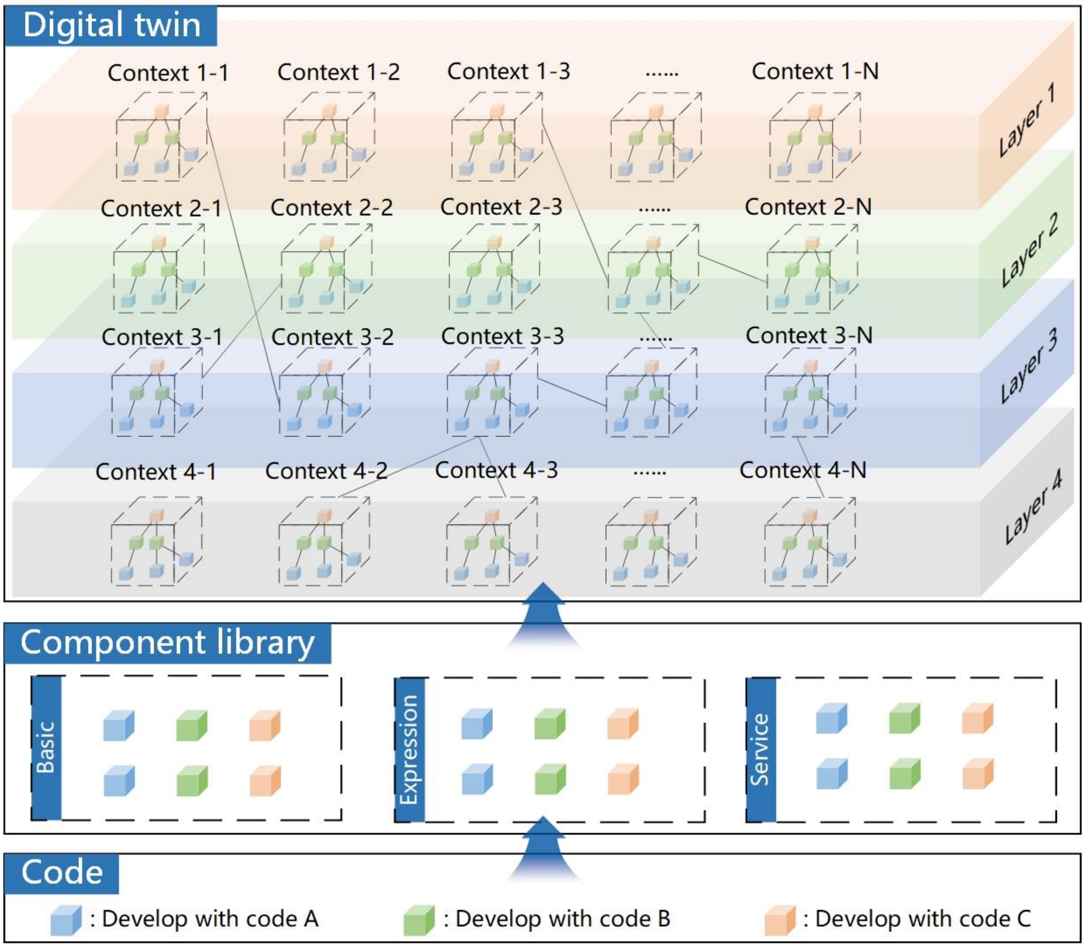


Figure 3. Modeling architecture of complex digital twins in 4C architecture

The implementation process of 4C architecture is as follows: Firstly, the complex digital twin is divided into several layers with different scales, determining the effective expression elements and negligible details at different layers. Then it is divided into different application contexts so that the divided simple digital twin only focuses on a specific function. After that, each simple digital twin is divided into several functional components according to the specific implementation process. The appropriate programming language or platform is used to develop and encapsulate the components, and the components with clear input, output, and association are formed. Finally, the components in the same context are integrated to construct the simple digital twin.

### 3.1 Division the complex digital twin in Composition and Context

**Composition:** The physical entities in complex digital twins usually belong to different contexts.

Moreover, the spatial span of different contexts is too large to be displayed at the same time, so the division of different scales is necessary. The complex digital twin can be divided into several layers into different scales, such as system, unit, part, or subpart. The elements that need to be expressed at different layers are not the same. System-layer contains all objects and their environment of the complex digital twin, the elements expressed in the system-layer digital twin include overall system operating conditions, operating rules, resource consumption, product output. The unit could be composed of several pieces of equipment to achieve a certain goal, the elements expressed in the unit-layer are mostly related to the functions, such as tasks progress, production efficiency, and product accuracy. The part-layer is the smallest unit of an executor of tasks, contains important movable parts related to the function of equipment, which usually needs to be monitored or controlled, such as the servo motor, the spindle. The part-layer express the elements are concentrated in key geometric and physical attributes such as size, the position coordinates and operating status of the equipment, the static or dynamic attributes. The subpart-layer can be the entities below the part size, which could make up the part, such as the gear to the spindle, or the microscopic effect that occurred on the part, such as the deformation, stress, fluid. Therefore, the modeling element and the data of the entire complex digital twin are selectively expressed at different layers, and these required data are obtained through sensors, controller of equipment, or additional edge controllers.

**Context:** Even at the same layer, the digital twin also covers multiple contexts, constructing a digital twin in a multi-scenario is not easy. Therefore, the division of multi-context is essential besides the division of different scales. After the layer division, the digital twins may still contain many physical entities, even a specific physical entity may have many application contexts. With the division of the context, the digital twin could be simplified accordingly. In a specific simple digital twin, usually only a single or limited context is considered, the parameters in the digital twin that do not belong to the current context can be hidden or regarded as constants, then the mathematical expression of the mechanism model is also simplified. After the division of the context, a digital twin could only focus on a single scale and a single context becomes many simple digital twins. For every simple digital twin, the specific implementation steps of it can be determined, including the input data and output results, structure, functional composition, implementation flow.

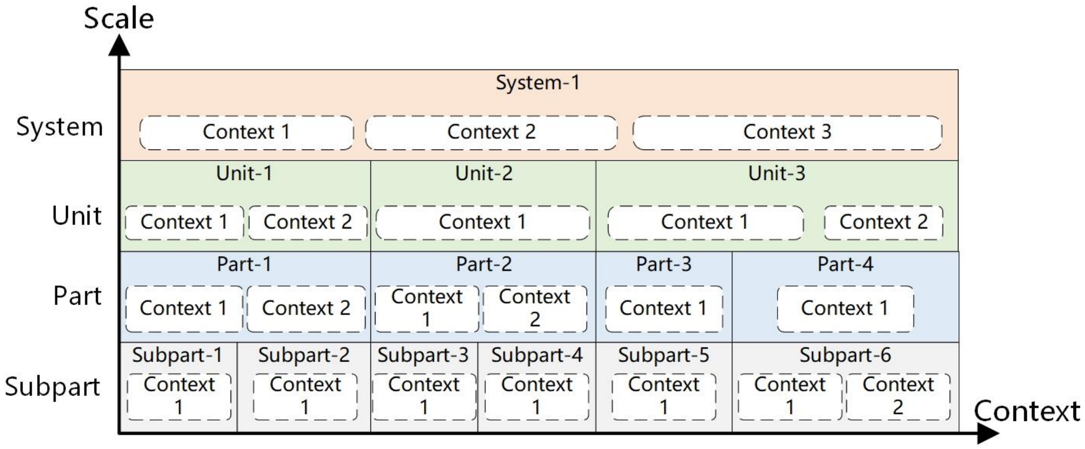


Figure 4. Division of different scales and context

The division of digital twins in scales and contexts is shown in Figure 4. Obviously, this two-tier division method turns a complex digital twin into many simple models. This division method enables the complex digital twin to have a certain degree of scalability. It allows different elements and data to be expressed on the corresponding layer and context, providing insight at different scales.

Dividing a complex digital twin into many simple digital twins also has some disadvantages. The physical entities, such as equipment, materials, person, are individuals that contain multi-dimensional information, but the division of different scales and contexts makes them into many discrete parts, and any of them only records part of its information and cannot represent the complete physical entity. Specifically, although the division of different scales allows data to be expressed on four layers, it also separates the attributes of some cross-scale physical entities into different layers. In addition, after the division of the multi-context, since only the parameters of a single context are considered, the accuracy of the mechanism model will be reduced. How to solve these problems will be discussed in Section 4. The following continues to discuss the implementation of simple digital twins.

### 3.2 Construct simple digital twin in Component and Code

After dividing the complex digital twin, the next step is to construct simple digital twins. The way to develop and implement the digital twin is diverse, and the software and development platform used by different developers in the implementation are also various, which also leads to poor portability and scalability of the digital twin. However, there are a lot of similar tasks in the process of constructing digital twins of different application contexts. And considering that software is the carrier of digital twins, the reuse of programs and codes is also important. To reduce the repetitive work in the development of the digital twin, similar tasks should be abstracted into serval independent standardized components according to the function, and then a component library of the digital twins is constructed, which can improve the portability and scalability effectively.

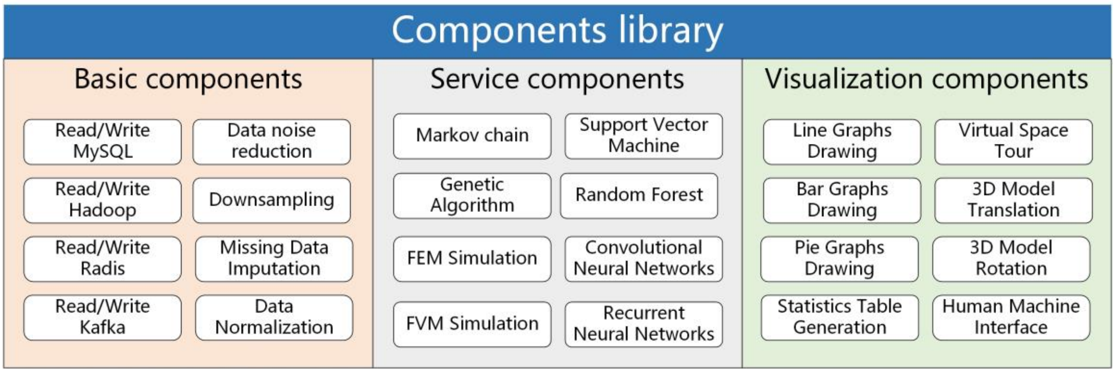


Figure 5. Classification of components

**Component:** As shown in Figure 5, the components of the digital twin can be divided into basic components, visualization components, and service components according to their functions. The basic components of the simple digital twins are mainly related to data, such as database operations (including reading and writing), data preprocessing. Visualization components are used to build the human-computer interaction interface and visualize the content on the interface. The function of the service components is to analyze the data collected in real-time based on the mechanism models and the data models and realize the functions of simulation, prediction, and optimization.

Although the kinds of components are various, they are basically organized according to two structures, as shown in Figure 6. The first structure is shown in Figure 6(a), which is centered on the operation of data, including “data input—data processing—data output”. Among them, data processing is the core of the required functions. Sources of data input include raw data in the database and output of other components. Data output includes other components in virtual space and entity in physical space. The second structure is shown in Figure 6(b), which is centered on the digital twin modeling, including “geometry—physics—behavior—rule”. Geometry is the CAD models of physical entities. Physics includes the physical attributes of physical entities, which can be simulated by software such as ANSYS. Behavior is the response of physical entities under the action of internal and external factors, which can be constructed by Markov chain, neural network and others. Rule is mined from data, often generated through machine learning or other algorithms.

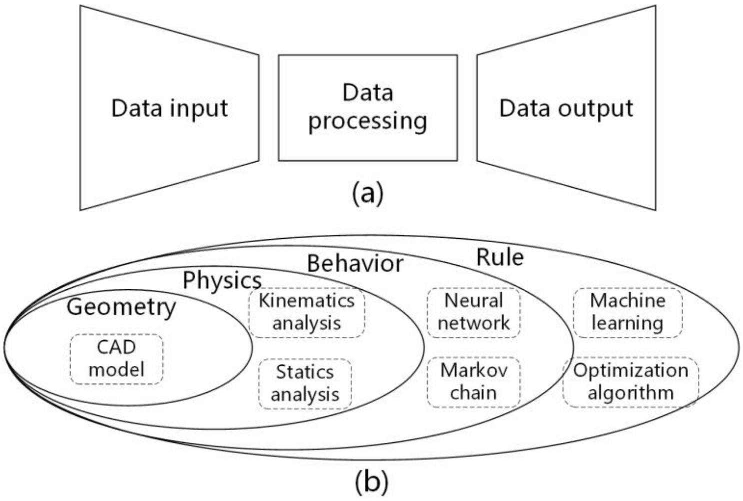


Figure 6. The structure of the component

The interfaces of changeable parameters in a component should be reserved in advance so that the parameters can be modified. The interfaces of the component mainly include name, ID, IP, composition, context, data input, data output, other changeable parameters, and description. Name is a generalization of the component function. ID is the unique identification in the digital twin system. IP is used to communicate with equipment within the network. Composition and context represent the layer and scenario of the simple digital twin in the 4C architecture respectively. Data input and data output are the interfaces for component interaction. Significantly, there are other changeable parameters in components, such as the acquisition frequency of the data acquisition component, the data dimension of the data analysis component, which also requires interfaces. In addition, an interface is reserved for recording a detailed description of component functions, which can be used to analyze the correlation of components through artificial intelligence technologies such as text recognition in the future.

**Code:** Code is the final step to develop a simple digital twin. The development process of the component usually depends on multiple programming languages or development platforms, so the code in a broad sense includes not only programming language codes, but also development platforms and software tools. The appropriate programming language or platform is chosen according to the actual requirements. According to the component structure described above, the code helps to develop the function of the component, such as data input, data processing, data output or the geometric shape, the physical attribute, the rule of relationship, the behavior, and finally encapsulate the interfaces of the component.

The componentization of functions enables the complex digital twin to have a certain degree of expansibility. In 4C architecture, components are the smallest functional units that make up the simple digital twin, so the evolution and update of digital twins are achieved by the creation, modification, and deletion of components, and the corresponding detailed operation processes are shown in Figure 7. When creating a new digital twin, according to the results of functional analysis and component division, determine in turn whether there are components that can realize the function in the component library. If this component exists, it can be used directly. If the component does not exist, the component should be developed and synchronized to the new component to the component library. Finally, the function of the digital twin is realized by connecting different types of components. When modifying a digital twin, offline or online functional testing of the components is required. Only the components that pass the test can replace the original components. At the same time, the modified components need to be updated to the component library, which is distinguished by adding the version number and description. When deleting a digital twin, because there may be functional dependencies between components, it is necessary to analyze the correlation of components to ensure that the deletion of the component will not affect the function of the system.

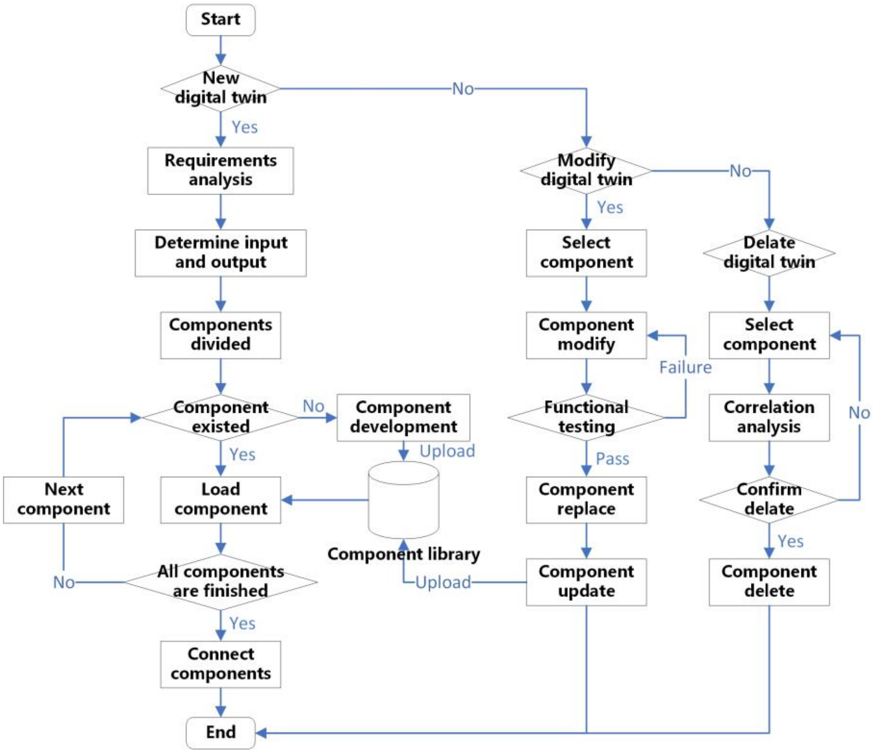


Figure 7. Digital twin creation, modification, and deletion process

## 4 Assemble the simple digital twins into a complex digital twin

The simple digital twins have been divided into specific scales and contexts in the above analysis. For assembling the complex digital twin, the information fusion, the multi-scale association, and multi-context interaction need to be considered. The ontology model provides a computer-understandable semantic description, which can be used as a container to fuse the related multi-dimensional information. The knowledge graph builds a semantic relation network between nodes, it is an effective tool for describing the scale associations. The multi-context interaction can be realized through behavior analysis and iteration of the related simple digital twins, but the specific interaction and iterative mode need to be selected according to the actual situation.

### 4.1 Information fusion based on ontology model

In Section 3, the data of the complex digital twin could be placed into different scales, for example in four layers: system, unit, part, and subpart. Then it is divided into different scenarios according to the context. After the division, the same physical entity may exist in several application contexts at the same time. The attributes of physical entities are scattered in many simple digital twins. However, in the complex context application, the data and attributes of simple digital twins need to be integrated into a whole database to achieve high-fidelity mapping of physical entities. Therefore, a container that can fuse information of simple digital twins in different scales and contexts is needed. Ontology is often used in the field of computer science. It is a conceptual model used to describe individuals (instances), classes (concepts), attributes, and relations. The ontology model provides a data structure describing physical entities, which can be used for information sharing and fusion. Extensible Markup Language, Resource Description Framework, and Web Ontology Language are common ontology languages. Protégé, XML Editor, and other development tools can be used to build the ontology models.

The structure of the ontology model of a complex digital twin is shown in Figure 8. The ontology-based data model may come from several simple digital twins. The data composition could be knowns as three categories, basic attributes, technical attributes, and state attributes. Basic attributes provide a general description of physical entities, such as name, ID, serial number. Technical attributes describe the physical structure and technical performance of physical entities, including geometric size, weight, load, and others. The basic attributes and technical attributes constitute the static data of the ontology model, and these data usually need to be entered manually. The state attributes are primarily related to the real-time data of equipment, and it is also the dynamic data of the ontology model, such as the overall operating status, position, angle, speed. Some data can be automatically obtained by data acquisition software; others may need to enter manually.

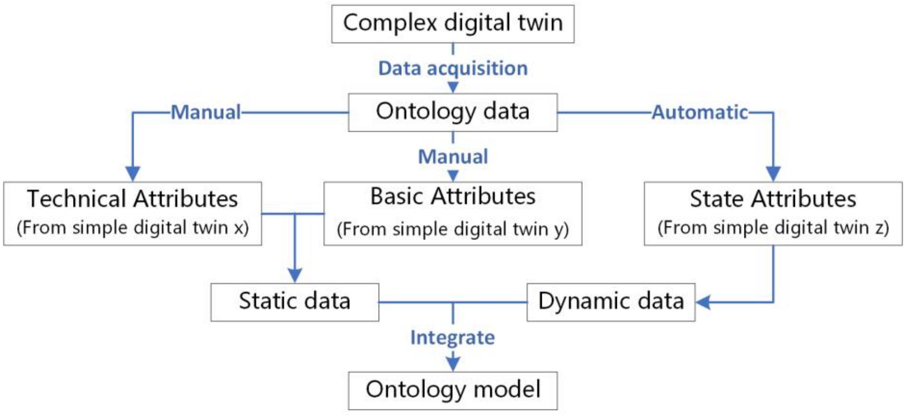
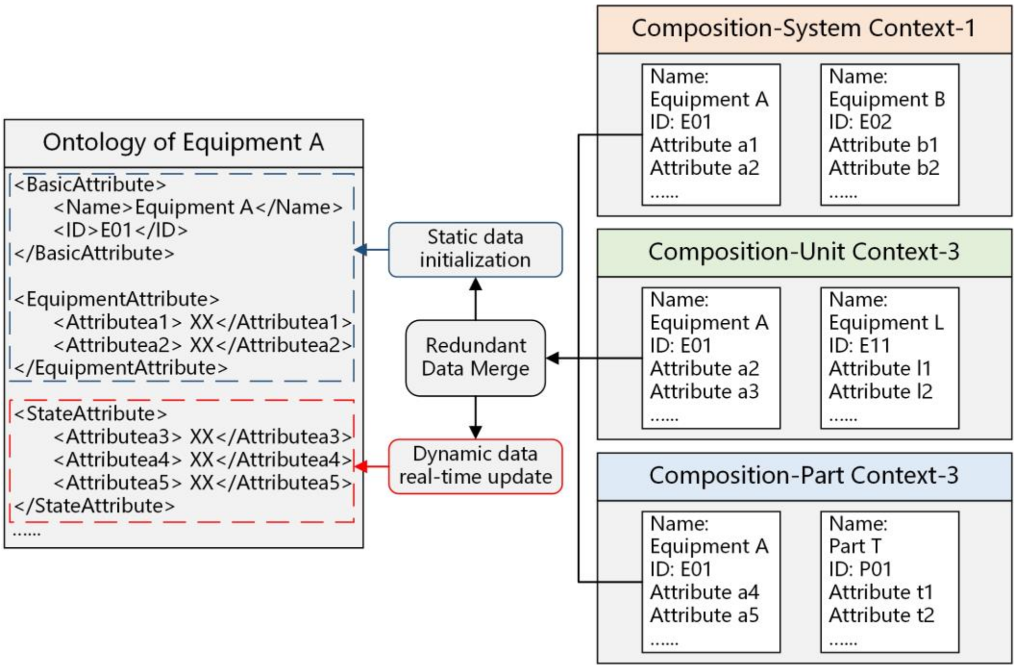


Figure 8. Data composition of ontology model

The ontology model of a complex digital twin contains the attributes of all related simple digital twins, which can be extensible or modifiable. In the process of information fusion, some data is redundant. For example, even in different simple digital twins, the name, ID of the same equipment should be the same, these same data only need to be recorded once. On the other hand, the data unique to each simple digital twin need to record in the ontology model of a complex digital twin. When the parameters of the simple digital twin are changed or a new one is created, the data of the ontology model can be updated accordingly.

The process of information fusion based on ontology is shown in Figure 9. However, the ontology model after information fusion has too many parameters, updating all parameters in real-time will take up a lot of computing resources, and not all parameters need to be updated in real-time. To solve this problem, static data and dynamic data are updated in different ways. Static data is initialized when the digital twin is created, and dynamic data is updated in real-time during the working of digital twins.

In this way, it can not only meet the application requirements but also reduce the load of software. In addition, these dynamic attributes should also be modifiable to adapt to new requirements in new application contexts.



Ontology of Equipment A
<BasicAttribute>
<Name>Equipment A</Name>
<ID>E01</ID>
</BasicAttribute>
<EquipmentAttribute>
<Attributea1> XX</Attributea1>
<Attributea2> XX</Attributea2>
</EquipmentAttribute>
<StateAttribute>
<Attributea3> XX</Attributea3>
<Attributea4> XX</Attributea4>
<Attributea5> XX</Attributea5>
</StateAttribute>
......
Static data initialization
Redundant Data Merge
Dynamic data real-time update
Composition-System Context-1
Name: Equipment A ID: E01 Attribute a1 Attribute a2 ......
Name: Equipment B ID: E02 Attribute b1 Attribute b2 ......
Composition-Unit Context-3
Name: Equipment A ID: E01 Attribute a2 Attribute a3 ......
Name: Equipment L ID: E11 Attribute l1 Attribute l2 ......
Composition-Part Context-3
Name: Equipment A ID: E01 Attribute a4 Attribute a5 ......
Name: Part T ID: P01 Attribute t1 Attribute t2 ......

Figure 9. Information fusion of the simple digital twins

### 4.2 Multi-scale association based on knowledge graph

Knowledge graph was first proposed by Google in 2012. It was applied to the information retrieval in the Google search engine, through the semantic retrieval of target information, the efficiency and quality of information retrieval can be improved. Lately, it is widely used in natural language processing, intelligence analysis, recommendation systems, and others. The essence of a knowledge graph is a semantic network that reveals the relation between physical entities. The construction of a knowledge graph usually requires multiple techniques related to knowledge, such as knowledge extraction, knowledge fusion. And triples are general representations of knowledge graphs that mainly include two basic forms: "Entity-Attribute-Value" and "Entity-Relation-Entity". In the graphical representation of a knowledge graph, nodes and lines are used to describe these triples. These nodes represent physical entities or attribute values, and the nodes are connected by lines, these lines are also called edges, which represent attributes or various semantic relations.

The complex digital twin should be able to display the entities in different scales as needed, and the model, the data, and the behavior could be adjusted accordingly as the perspective changes. A knowledge graph can be used to achieve the above function. In Figure 10. the structure of the knowledge graph can be thought into two categories, class and instance. The class includes the type, subtype, and subordination of nodes. Here, the type is consistent with the layers of digital twin such as system, unit, part, and subpart. The subtype is details of the objects, such as spindle, machine tool, production unit. In the subordination, a parent node usually corresponds to several child nodes, and a child node corresponds to only one parent node, so the parent node is recorded to represent the subordination.

Instance refers to a specific object in the physical space, it has two categories, attributes, and relations. Attributes include ID, size, weight, and other basic attributes of an object. Relations contain the IDs of nodes associated with the current node and the description of the relations between these nodes. In the above structure, the class and the attributes of instance are supplied in the ontology model, then the relations of these simple digital twins can be determined according to the knowledge graph.

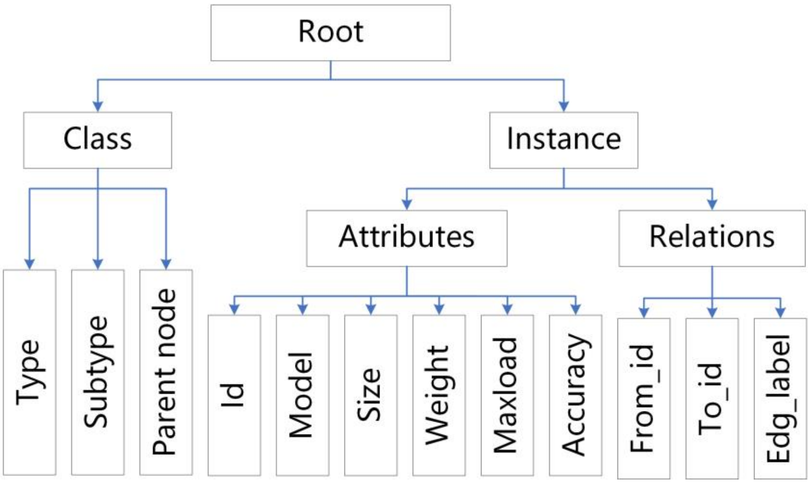
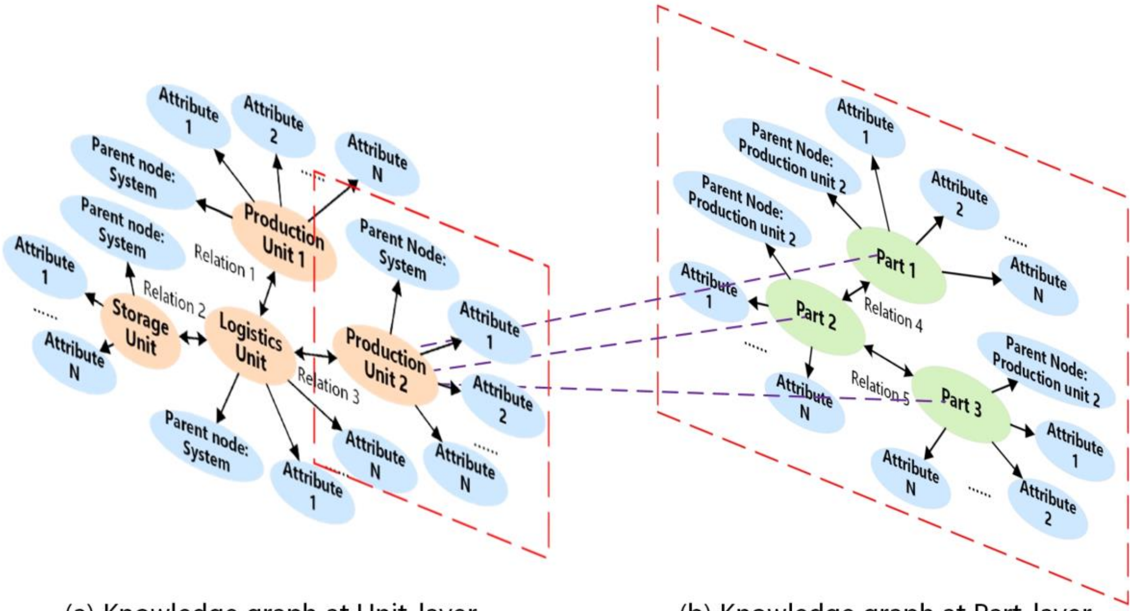


Figure 10. The data structure of the knowledge graph

Figure 11 shows an example of the knowledge graph in a complex digital twin at different scales. The class in structure records the scale of the digital twin and its parent node, so a knowledge graph can provide a bridge between digital twins in different scales. Besides, the data displayed by the complex digital twin can be selected according to the following strategy: if all child nodes of a node are included in the scenario, the attributes and relations of these child nodes are ignored. Conversely, if a node has only part of its child nodes included in the scenario, then the attributes and relations of the parent node are ignored.



(a) Knowledge graph at Unit-layer

(b) Knowledge graph at Part-layer

Figure 11. The scale association in a knowledge graph

### 4.3 Multi-context interaction based on behavior analysis and iteration

The actual physical process is usually the result of the interaction of multi-context. In a complex digital twin system, the interaction of contexts can make the virtual entities more truly reflect the status of the physical entities. As shown in Figure 12, the interaction of multi-context digital twins can be summarized into three types: superposition, transmission, and iteration. Superposition is the direct addition of digital twin outputs. For independent digital twins, since there is no interaction between them, the interaction of contexts is the direct addition of the output results. Transmission means using the output of one digital twin as the input of another digital twin, which is suitable for one-way calculation. For example, if the calculation of context A needs the output of context B, context B needs to be calculated first, and then use the result of B as the input of A to get the final result. Iteration means the output of two digital twins affected by each other. Context A and Context B interrelate, which is suitable for two-way interaction. In this case, the output of context A affects context B, and the output of B also affects A. When two contexts interact, the output of A is calculated first and used as the input of B, then use the output of B as the input of A. Repeat the above process until the final results converge.

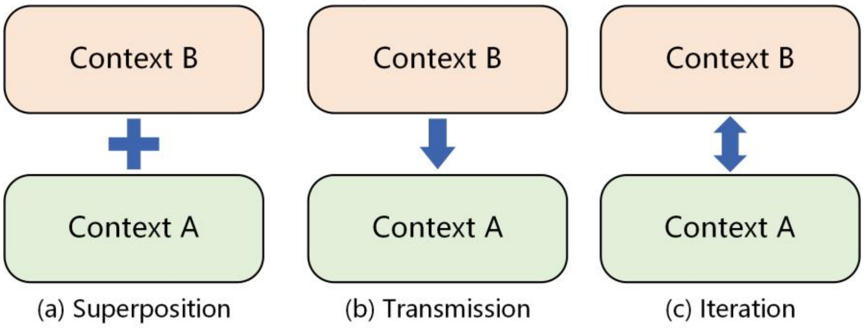


Figure 12. The interaction of multi-context digital twins

The key to realizing the multi-context simulation of the complex digital twins is to find the associate intermediate variables of digital twins. List the input, output of digital twins, and the output of all service components, compare and analyze the variables of different digital twins to find the associated intermediate variables. If the associated intermediate variables are absent, the superposition is selected to update the output of the two digital twins. If the output of one digital twin is the intermediate variable of another digital twin, the transmission way is selected to update the output of digital twins. If the output of one digital twin is the intermediate variable of another and vice versa, the interactions of digital twins are realized by iteration. To prevent the update of the digital twin from falling into an infinite loop, it is necessary to set the termination condition of the iteration. For instance, the termination condition could be the error of adjacent outputs is less than 0.01(1%) or the number of iterations exceeds a certain number.

## 5 Conclusion

To set up the complex digital twin in multi-scale, multi-dimensional information and multi-scenarios, a novel modeling method is proposed in this article.

The complex digital twin is firstly divided into several achievable simple digital twins by composition, context, component, and code in 4C architecture. Then assemble the simple digital twins to construct a complex digital twin through information fusion, multi-scale association, and multi-context interaction. The interesting results of the article could be summarized as follows,

1) Four important characteristics of complex digital twins were discussed. The connotation of complex digital twins needs to satisfy scalability, interoperability, expansibility, and fidelity. The requirement of a complex digital twin is given as follows, it should be able to express information at different scales, simple digital twins in system can interact with each other, components in digital twins should be easy to create or replace, the complex digital twin should also be as close to the physical entity as possible.

2) The first step is to divide a complex digital twin into several achievable simple digital twins.

4C modeling architecture (Composition-Context-Component-Code) is proposed to divide a complex digital twin into several simple digital twins. The composition constrains the elements of the digital twin on a limited scale. The context allows the digital twin to focus on the specific behavior and output. Such division turns a complex system with multiple scales and multiple contexts into several simple digital twins that can be implemented. Furthermore, the components are constructed based on the standard modular and reusing ability. It can quickly build the required digital twins. Afterward, the code could be developed based on the standard interface for realizing the simple digital twins.

3) Secondly, the ontology, knowledge graph, and multi-context behavior analysis are introduced to assemble simple digital twins into a complex digital twin. The ontology model integrates the basic attribute, technical attribute, and status attributes coming from the simple digital twin. The effective methods to data update are necessary for realizes the information fusion as static data and dynamic data, which means that static data is only updated when the digital twin is initialized, and the dynamic data is updated in real-time during the working of digital twins. The knowledge graph records the necessary attributes of a physical entity and sets the relations between the simple digital twins. It can also enable the expression of a complex digital twin adjusted according to the specific contexts. Next, three types of interaction between digital twins are studied, which can improve the accuracy of the behaviors of the complex digital twin based on the existence of the intermediate variables.

The modeling method proposed in this paper provides a reasonable methodology and operable guidance for the modeling of a complex digital twin in multi-scale and multidimensional information fusion and multi-context. The implementable of the processing methods on the complex digital twin will be explained in part II and also provides the support for prediction, optimization based on the complex digital twin.

## Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 52175456), NSAF U1830110 and the Fundamental Research Funds for the Central Universities ZYGX2019J032.

## References

[1] M. Grieves, Digital twin: manufacturing excellence through virtual factory replication, White paper, Melbourne: US Florida Technology, 2014.

[2] E. Glaessgen and D. Stargel, “The digital twin paradigm for future NASA and U.S. Air Force vehicles,” in Proc. 53rd IAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf., 2012. [Online]. Available: https://arc.aiaa.org/doi/pdf/10.2514/6.2012-1818.

[3] D. Liu, H. Huang, B. Wang, T. Zhou, S. Luo, Operation paradigm for remanufacturing shop-floor based on digital twin, Computer Integrated Manufacturing Systems 25(06) (2019) 1515-1527.

[4] M. Grieves, J. Vickers. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex system//Trans-disciplinary Perspectives on Complex Systems. Berlin, Germany: Springer-Verlag, 2017

[5] F. Tao, M. Zhang, Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing, IEEE Access 5 (2017) 20418-20427.

[6] C. Zhuang, J. Liu, H. Xiong, X. Ding, S. Liu, G. Weng, Connection, architecture and trends of product digital twin, Computer Integrated Manufacturing Systems 23(04) (2017) 753-768.

[7] K.M. Alam, A. El Saddik, C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems, IEEE Access, 5 (2017) 2050-2062.

[8] D. Guo, J. Bao, G. Shi, Q. Wan, X. Sun, H. Weng, Research on Modeling of Aerospace Structural Parts Manufacturing Workshop Based on Digital Twin, Journal of Donghua University (Natural Science) 44(04) (2018) 578-585+607.

[9] J. Guo, H. Hong, K. Zhong, X. Liu, Y. Guo, Production Management and Control Method of Aerospace Manufacturing Workshops Based on Digital Twin, China Mechanical Engineering 31(07) (2020) 808-814.

[10] R. Soderberg, K. Warmefjord, J.S. Carlson, L. Lindkvist, Toward a Digital Twin for real-time geometry assurance in individualized production, Cirp Annals-Manufacturing Technology 66(1) (2017) 137-140.

[11] C. Zhuang, J. Liu, H. Xiong, Digital twin-based smart production management and control framework for the complex product assembly shop-floor, International Journal of Advanced Manufacturing Technology 96(1-4) (2018) 1149-1163.

[12] B. Bjornsson, C. Borrebaeck, N. Elander, T. Gasslander, D.R. Gawel, M. Gustafsson, R. Jornsten, E.J. Lee, X. Li, S. Lilja, D. Martinez-Enguita, A. Matussek, P. Sandstrom, S. Schafer, M. Stenmarker, X.F. Sun, O. Sysoev, H. Zhang, M. Benson, C. Swedish Digital Twin, Digital twins to personalize medicine, Genome Medicine 12(1) (2019)

[13] K. Bruynseels, F.S. de Sio, J. van den Hoven, Digital Twins in Health Care: Ethical Implications of an Emerging Engineering Paradigm, Frontiers in Genetics 9 (2018).

[14] P. Xu, W. Chen, L. Liao, Z. Zhang, Y. Feng, Research on digital workshop of ship pipe machining based on digital twin, Ship Science and Technology 41(15) (2019) 139-144.

[15 F. Dembski, U. Woessner, M. Letzgus, M. Ruddat, C. Yamu, Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany, Sustainability 12(6) (2020).

[16] C. Fan, C. Zhang, A. Yahja, A. Mostafavi, Disaster City Digital Twin: A vision for integrating artificial and human intelligence for disaster management, International Journal of Information Management 56 (2021).

[17] H.R. Zhao, J.H. Liu, H. Xiong, C.B. Zhuang, T. Miao, J.S. Liu, B. Wang. 3D visualization real-time monitoring method for digital twin workshop. Computer Integrated Manufacturing System, 2019, 25(6): 1432-1443.

[18] T.G. Ritto, F.A. Rochinha, Digital twin, physics-based model, and machine learning applied to damage detection in structures, Mechanical Systems and Signal Processing, 155 (2021).

[19] Q. Lv, R. Zhang, X. Sun, Y. Lu, J. Bao, A digital twin-driven human-robot collaborative assembly approach in the wake of COVID-19, J. Manuf. Syst. 60 (2021) 837-851.

[20] X.N. Lai, S. Wang, Z.G. Guo, C. Zhang, W. Sun, X.G. Song, Designing a Shape-Performance Integrated Digital Twin Based on Multiple Models and Dynamic Data: A Boom Crane Example, Journal of Mechanical Design, 143 (2021)

[21] R. Rosen, G. von Wichert, G. Lo, K.D. Bettenhausen, About The Importance of Autonomy and Digital Twins for the Future of Manufacturing, Ifac Papersonline 48(3) (2015) 567-572.

[22] S.L. Wang, Y.K. Wang, B. Yang, S.B. Wang, Fog manufacturing: New paradigm of industrial internet manufacturing based on hierarchical digital twin. Computer Integrated Manufacturing Systems, 2019, 25(12): 3070-3080

[23] Y.H. Pan, T. Qu, N.Q. Wu, M. Khalgui, G.Q. Huang, Digital Twin Based Real-time Production Logistics Synchronization System in a Multi-level Computing Architecture, J. Manuf. Syst., 58 (2021) 246-260.

[24] Q. Bao, G. Zhao, Y. Yu, S. Dai, W. Wang, The ontology-based modeling and evolution of digital twin for assembly workshop, International Journal of Advanced Manufacturing Technology (2021) 1-17.

[25] S. Dai, G. Zhao, Y. Yu, P. Zheng, Q. Bao, W. Wang, Ontology-based information modeling method for digital twin creation of as-fabricated machining parts, Robotics and Computer-Integrated Manufacturing 72 (2021).

[26] S. Singh, E. Shehab, N. Higgins, K. Fowler, D. Reynolds, J.A. Erkoyuncu, P. Gadd, Data management for developing digital twin ontology model, Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture, (2020).

[27] Y.Q. Lu, X. Xu, Resource virtualization: A core technology for developing cyber-physical production systems, J. Manuf. Syst., 47 (2018) 128-140

[28] S. Liu, Y. Lu, J. Li, D. Song, X. Sun, J. Bao, Multi-scale evolution mechanism and knowledge construction of a digital twin mimic model, Robotics and Computer-Integrated Manufacturing 71 (2021).

[29] X. Sun, R. Zhang, S. Liu, Q. Lv, J. Bao, J. Li, A digital twin-driven human-robot collaborative assembly-commissioning method for complex products, International Journal of Advanced Manufacturing Technology, (2021).

[30] B. Schleich, N. Anwer, L. Mathieu, S. Wartzack, Shaping the digital twin for design and production engineering, CIRP Ann-Manuf. Technol. 66(1) (2017) 141-144.